CLOUD BASED N-DIMENSIONAL WEATHER FORECAST VISUALIZATION TOOL WITH IMAGE ANALYSIS CAPABILITIES

M. Laka-Iñurrategi [a], I. Alberdi [a], K. Alonso [b], M. Quartulli [a]

[a] Vicomteh-IK4, Mikeletegi pasealekua 57, 20009 Donostia-San Sebastián, Spain
{mlaka, ialberdi, mquartulli}@vicomtech.org
[b] The German Aerospace Center (DLR), Oberpfaffenhofen, D-82234 Wessling, Germany
kevin.alonsogonzalez@dlr.de

KEY WORDS: Meteorology, Image Processing, Visualization, GIS

ABSTRACT:

Until recently the majority of data analysis and visualization tools were desktop applications that demanded high requirement hardware to carry out those processes. However, nowadays there is a trend to evolve this kind of applications to service based solutions that can be accessed remotely. Considering the implications that the weather has in the health and the safety of the human beings, authorities require a further knowledge of the weather forecasts and their impacts but they have difficulties to properly understand the raw forecasts since they usually are not experts in the field of meteorology. For this purpose, we have designed and implemented a framework that permits a remote access to weather forecasts. With this tool, the practitioners can access, visualise and interact with the data from a web browser. Furthermore, it contains an image and numeric analysis module that permits the generation of new information what is helpful in decision making processes.

1. INTRODUCTION

Meteorology has become a field of knowledge that government and agencies are increasingly demanding as an essential information source in their decision making processes. Nevertheless in the majority of cases these practitioners are not experts in meteorology so they fail to properly interpret the predictions and products that meteorology agencies provide them. Therefore, the authorities need specific tools that enhance their decision making processes by facilitating the understanding and management of meteorological observation and prediction products [1].

The major requirement for a tool that manages meteorological observations and forecasts consist on handling spatial data. Somehow these tools are a specific use case of generic Geographic Information Systems that need to manage meteorological data (spatial data) and offer specific analysis and visualization capabilities of the field of meteorology.

The traditional desktop GIS programs are evolving into distributed applications in order to be able to interact with online geospatial resources. As a result, client-server or browser-server based distributed GIS applications have been introduced to fill the gap, and the GIS companies and research groups have developed their own spatial databases along with various data access and manipulation tools [2]. The principal trend is to transform the GIS applications towards systems that can be accessed as a service rather than to run them on the individual computers [3].

Geospatial applications have to face several challenges related to the size of spatial data, the heterogeneity of this data, the complexity of loading it into the clients and the time required for analysing big and complex data [2]. However, there exist several tools that offer a GIS service with a whole chain behind which includes the data storage and visualization and analysis capabilities at server side.

But regarding meteorological applications, there is lack of frameworks that offer web-based interaction capabilities with meteorological raw data at server side.

In this regard, this paper describes a framework designed for the specific requirements that a remote meteorological data management application demands. It is a tool based on web and cloud technologies for the visualization of weather forecasts. The tool stores the weather forecasts in a dynamically scalable repository and offers visualizations on top of a web mapping client. The user can choose the parameters to visualize, create personal combinations and carry out some analysis. It is based on open source geospatial servers and web mapping clients and uses OGC standards for data sharing and service providing. We introduced some improvements in our framework that are explained in the next sections.

This paper is organized as follows. The section 2 describes the features of the data format used for storing weather forecasts that the tool has to handle. The implementation of the framework is described in section 3 which includes the architecture and the extensions introduced for expanding the visualization capabilities and improving the performance. In section 4 the analysis module is described and in section 5 the results of the achieved framework are given. Finally, some conclusion remarks are presented in section 6.

2. DATA

The aimed framework has to handle weather forecasts as source data. In particular, weather forecasts produced by widely employed Weather Research and Forecasting (WRF) Model [4]. It is a mesoscale numerical weather prediction (NWP) system for the simulation and prediction of the atmosphere. Therefore, the output consists on a set of weather-related parameters for each of the points of a grid domain where the forecasting is processed. These output parameters, by default, are written in

NetCDF[1] data format which is a standard data form defined for the creation, access and sharing of N-dimensional array oriented scientific data. Thus, the framework will have to handle weather parameters stored in a set of NetCDF files.

In this case, the forecast for the next two days is generated every day. A unique NetCDF file is generated per each hour of these two days containing all the parameters that define the forecasted atmospheric situation at this time. Therefore, each day, the system will receive 48 NetCDF forecast files.

The forecasting model generates a lot of parameters such as the surface temperature, surface pressure, convective precipitations and so on. However, there are other parameters required by the users such as thermal comfort or wind chill that have to be processed afterwards.

3. SYSTEM DESCRIPTION

The visualization tool described in this work is a web-based framework that generates graphic visualizations of forecasted meteorological parameters. It also offers image and alphanumeric analysis capabilities over these forecasted parameters. All these operations are carried out in real time as a major requirement for web technologies.

The architecture of the describing framework is based in a client-server distributed application structure. The prediction files are stored and managed at server side where the calls of the clients are attended and all the analysis and visualization operations are located. At the client side, a web interface helps users to send personalized visualization and analysis requests to the server.

3.1 Implementation

The implementation of the visualization and analysis tool is shown in Figure 1.

The server side is mainly based on Mapserver[2] development environment and ncWMS[3] web map service. Mapserver is an Open Source platform for publishing spatial data that provides capabilities to build interactive mapping applications to the web. Mapserver supports several data input formats "natively" and many more if it is compiled with the open source libraries GDAL and OGR. Mapserver uses GDAL for reading of netCDF files, but since GDAL itself does not support all versions of netCDF files it is recommended to convert files into GeoTiff format first[2]. Since the server will receive 48 netCDF files each day, converting all the files would worsen the performance of the whole system. This fact makes necessary the use of ncWMS.

ncWMS is a Web Map Service for geospatial data is stored in CF-compliant netCDF files. ncWMS itself can provide geo-registered map images from a netCDF file repository as a reply of a WMS request. Thus, in this architecture, Mapserver receives a visualization request from the client that redirects to ncWMS as a WMS request in order to obtain map images from netCDF files.

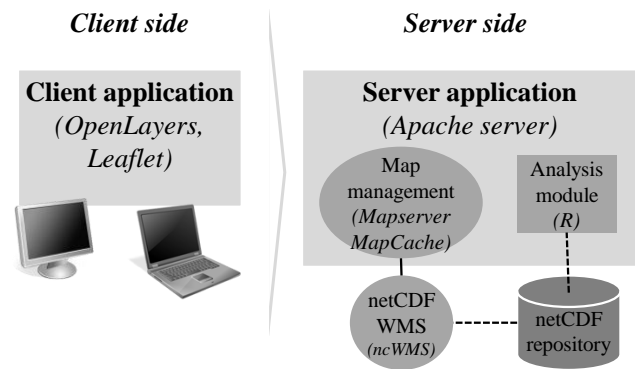But Mapserver offers more capabilities than just generating map

Figure 1. Architecture of the framework.

images from data sources. In this system, the use of Mapserver has speed up the access to the map images thanks to the included MapCache server. With MapCache the tiles generated for the first time are stored in a cache so that the time of having to regenerate visualizations is avoided for the subsequent calls. In addition to this, Mapserver maintains opened the possibilities to add further data sources that could be managed directly with Mapserver that ncWMS can not.

At the client side, the client application includes both OpenLayers[4] and Leaflet[5] javascript libraries for displaying forecasted parameters as map data in web browsers. Both libraries are included in order to compare their performance under the requirements of this framework.

Our approach has extended the capabilities that ncWMS and selected javascript libraries offer. In case of ncWMS vector-barb and contour image map generation capability has been added and in case of OpenLayers and Leaflet visualization improvements for softening the transition between layers.

3.2 Extending visualization capabilities

The proper interpretation of the meteorological parameters forecasted by a numeric weather prediction model requires specific visualizations that professionals are used to manage. The release version 1.1 of ncWMS does not generate contour visualization nor vector barbs whereas in weather forecasting, the ways of representing data such as wind-barbs or iso-pressure lines were standardized long ago.

ncWMS do permit vector visualization but it does not include barb symbols. The algorithm used by ncWMS for extracting data and generating images follows 6 steps [5]. The first four are related to the definition of the transformation from the gridded source data to the target coordinate reference system. The fifth step comprises the extraction of the information from the source grid and the sixth, converting the extracted data to an image. This last step was extended modifying the server source code in order to be able of processing new visualization layers. The layer type to represent is specified by the LAYER parameter defined in the request procedure of the WMS standard. The barb visualization is implemented directly over the ncWMS server code adding colour and barbs to the vector visualization of ncWMS (Figure 2). In order to be able to use

---

[1] http://www.unidata.ucar.edu/software/netcdf/

[2] http://mapserver.org

[3] http://www.resc.rdg.ac.uk/trac/ncWMS/

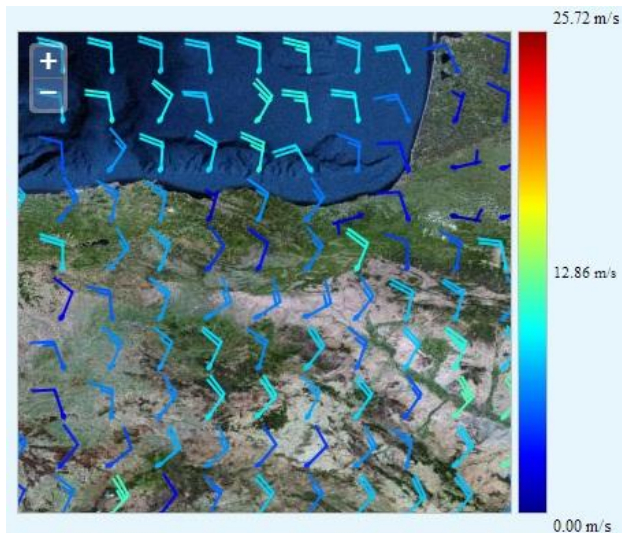[4] http://openlayers.org/

[5] http://leafletjs.com/

Figure 2. The visualization of extended wind-barbs.

this visualization it must be set in the STYLE parameter of the WMS request as "barb/rainbow".

On the other hand, the isoline visualization is carried out in the analysis module of the framework. ncWMS does not have contour generation features, but R scripting language does. Therefore the contour image generation is carried out in the analysis module and afterwards the image is encoded in the grid so that ncWMS draws a isoline map image with transparencies. This can be appreciated in Figure 3 where the blue lines represent the isolines for the pressure parameter.

3.3 Operation improvements

The framework requires some improvements at the implementation stage with the aim of adjusting it to the general requirements of the application.

At client side, both javascript libraries would initially consider a layer for each parameter at a time. Therefore, if a parameter is been visualized and its time evolution wanted to be assessed, the system will have to charge one layer after the other. With any adaptations it would cause flashy transitions from one visualization to the next. In order to solve it and achieve soft transitions, we defined two height levels for the layers at client side. This manner, the parameters are charged alternatively in each of the levels and the previous layer is not deleted until the new one is totally charged.

At server side, in order to extend Mapcache features to the map images that ncWMS generates, an authomatic mapping service has been included. When a netCDF file is received this service is activated for writing the description file of MapCache. This file defines the correspondence between the calls that the client does to Mapserver and the WMS calls that Mapserver will have to adapt and redirect to ncWMS. The calls of the client just contain the parameter that is wanted to visualize, the date, the time and the height. Thanks to the description file written by the service, MapCache is capable of binding the request of the client with the standard WMS call to ncWMS.

4. ANALYSIS

The purpose of this tool is not to be a mere visualization tool. The user could need to interact with the data and analyse it in order to draw conclusions on the forecasted weather and its implications. Nonetheless, there exist some browser GIS clients that provide viewing, editing and analysing of spatial data. Thus, the aim is not to develop a completely new spatial data analysis tool but to establish a framework that permits the analysis of meteorological data through a web client and remotely. Moreover, most of those browser GIS clients cannot manage netCDF filetypes directly, a must for applications that need to handle meteorological parameters. And additionally, most of those desktop applications that manage netCDFs, based their analysis capabilities just in the amount of different kind of visualizations they offer.

The users of this framework need to manage the data in order to extract new knowledge, and for that purpose data analysis capabilities are required. With this aim spatial analysis module based on R scripting language is accessible from the application server in the proposed framework.

R is a widely used software environment for statistical computing and graphics with a wide variety of functions for analysing and processing both geographic data sets and raster data. In particular, it can read and write netCDF file formats. Therefore, it can be used to manage the parameters in netCDF files as numeric variables or raster layers. For raster layer consideration, R itself converts the variables to RGB images before.

In this particular framework, on one hand, the R-based analysis module generates new variables (such as thermal comfort and wind chill) as a result of numeric processing of the original variables. On the other hand, it processes the meteorological parameters graphically to obtain the boundaries between specified values of a parameter what is helpful for highlighting regions on a map. A use case of this feature is the generation of contour maps for the visualization of the pressure.

The modularity of the architecture as well as the consistent input/output interfaces have permitted the integration of this image processing unit based on R scripting language and it would also permit the addition of new functionalities that could be based in other programming languages or technologies.

5. RESULTS

The aim of this framework was to achieve a web-based application that could offer graphic visualizations in real time of weather forecasts. Additionally the framework had to offer analysis capabilities that would help to extract new information from the source data.

The system implemented in the initial stages as SaaS (Software as a Service) cloud layer can easily install in production in any common cloud HaaS (Hardware as a Service) layer systems available in the market (AWS-EC2, GCE…). A SaaS application allows lightweight clients, avoiding computational expensive processing tasks because data reading management, fusion, processing and the generation of visualizations are all done in the server side. Thus, the system user only needs a HTML5 compatible browser to be able to access and use the tool.

At client side, both Leaflet and OpenLayers libraries were implemented for displaying the map images in web browsers. Both technologies met the requirements of this framework. Nonetheless, Leaflet showed a better performance and a quicker response to the client requests. For future developments of the tool the access from mobile devices will be required and nowadays Leaflet shows a better performance for that purpose. However, Openlayers is a consolidated framework with better documentation and community activity that must be taken into account.

With this framework, combined visualizations can be offered to the user. In Figure 3 the visualization of three parameters using OpenLayers library is shown. The background colour map is the forecasted temperature at surface in Kelvin, the blue lines are the processed isobars and the arrows the wind direction and intensity at surface.

A selection feature was also implemented which results are shown in Figure 4. In this case, a specific temperature range defined by the user is highlighted making use of Leaflet library. The colour regions represent the locations where the temperature is forecasted to be under 273 Kelvin and the red lines refer to the main roads of the area.
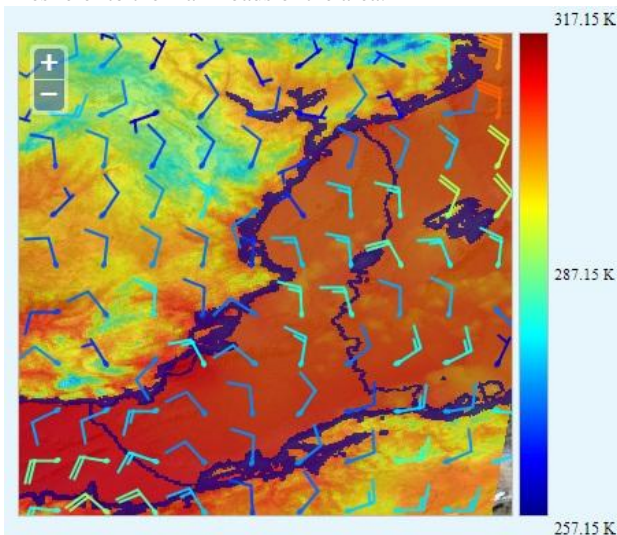

Figure 3. Visualization the forecasted temperature, pressure and wind at surface using OpenLayers library.
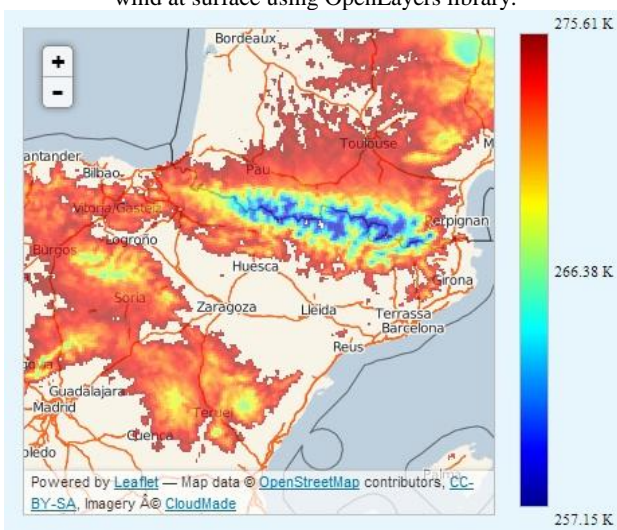

Figure 4. Visualization of the required temperature range using Leaflet library.

6. CONCLUSIONS

Offering access, manipulation and interaction of spatial data as a service is being increasingly demanded since it solves the issues of high hardware requirement and great amount of data maintenance. Moreover, it also provides universal access to the data and the features that the tools offer.

In this work, a framework for offering weather forecasts remotely has been described. This tool permits the access, interaction, visualization and new knowledge generation of meteorological parameters that weather forecast models provide.

The proposed architecture is mainly based in Mapserver and ncWMS at server side. Making use of ncWMS has avoided the difficulties that Mapserver has handling the NetCDF files that store the weather forecasts. The use of Mapserver has provided performance improvements mainly based on the tile caching that MapCache offers for speeding up the access to the layers that could not be possible to achieve only with ncWMS as image map server. It must be remarked that an analysis module has been also introduced. This module is based in R-scripting language but the modularity of the designed architecture will permit in future the integration of other technologies if they are required for the analysis.

At client side, both OpenLayers and Leaflet have been implemented with similar results for the required visualizations. However, Leaflet has proven to be faster charging the image maps than OpenLayers.

The novelty of the proposed architecture is that independent spatial related and web-based technologies have been implemented in a unique framework for developing a complete tool that manages complex spatial data as weather forecasts. Moreover the modularity of the framework permits seamless integration of data processing algorithms and new visualizations that enhance the capabilities of the base technologies.

7. REFERENCES

[1] Xiaojuan Sun et. al. Development of a Web-based visualization platform for climate research using Google Earth. *Computers & Geosciences 47,* pp. 160-168, 2012.

[2] Jeberson Retna Raj, R.; Sasipraba, T. Disaster management system based on GIS web services. In *Recent Advances in Space Technology Services and Climate Change (RSTSCC)*, pp. 252-261, 2010.

[3] Bian Wu; Xincai Wu; Jian Huang. Geospatial Data Services within Cloud Computing Environment. In *Audio Language and Image Processing (ICALIP)*, pp. 1577-1584, 2010.

[4] Michalakes, J., J. Dudhia, D.O. Gill (more): The Weather Research and Forecast Model: software architecture and performance. *11th Workshop on the Use of High Performance Computing in Meteorology*, 2004.

[5] J.D. Blower; A.L. Gemmell; G.H. Griffiths; K. Haines; A. Santokhee, X. Yang. A Web Map Service implementation for the visualization of multidimensional gridded environmental data. In *Environmental Modelling & Software 47*, pp. 218-224, 2013.
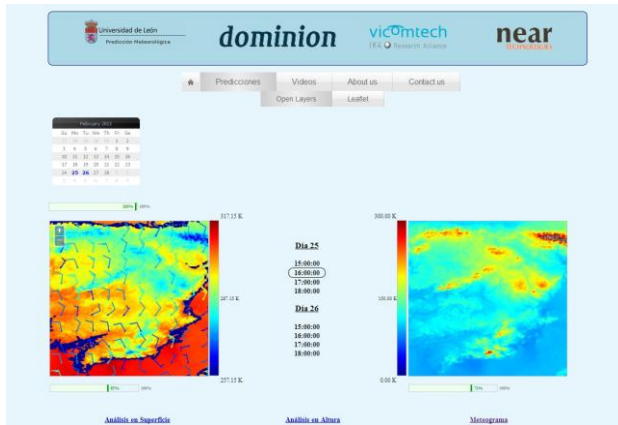
Figure 5. Client interface of the framework.